

Python Robotics Version A Project

|  |  |  |  |
| --- | --- | --- | --- |
| **Faculty** | Information Technology | | |
| **Module Name** | Python Robotics | **Module Code** | ITRCA0/ITMRA0/  ITMR003 |
| **Project Number** | A | **Copy Editor** | Ms Nicole Stern |
| **Student Name** | Sopumelela | **Date Submitted** | 30/08/2024 |
| **Student Surname** | Sandekela | **Student No** | CON-1475940-C5L6 |
| **Marker Name** |  | **Date Marked** |  |
| **Mark** | /140 | **Percentage** | % |
| **Moderator Name** |  | **Date Moderated** |  |
| **Mark** | /140 | **Percentage** | % |

|  |  |  |  |
| --- | --- | --- | --- |
| **Question Number** | | **Total** | **Mark Achieved** |
| Question 1 | | 71 |  |
| Question 2 | | 69 |  |
| **Lecturer** |  | **Total Mark** |  |

# ITRCA0 – Version A **Python Robotics**– Project Specification 2024 | V3.0 Page 1 of 4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirements** | | **Marks** | | **Deductions** |
| Robot mobility in  multiple directions | Robot is automated and moves in  different directions. | 10 | |  |
| Robot covers a good distance | The robot is far from its initial starting point. It just chooses a direction and moves in that direction for at least 15cm. | 5 | |  |
| Robot moves in a  random direction | There is no specific pattern to the  robot’s movements. | 5 | |  |
| Robot reverses  instructions | The robot runs all the processed  instructions in reverse. | 15 | |  |
| Robot emergency memory | Robot memory operates with no mistakes and when the robot shuts down unexpectedly it saves the last processed instruction and where it  stopped. | 14 | |  |
| Indicating light | A drawing of a fully labelled circuit to  integrate with the GoPiGo circuit. | 9 | 12 |  |
| All the required components to integrate  the new modification. | 3 |
| Code to turn on the light | Code that works with the GoPiGo to show the indicating light just before the  GoPiGo is about to turn | 10 | |  |
| **Total** | | **71** | |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirements** | | **Marks** | | **Deductions** |
| Adding AI functionality | The functionalities to be added are named. | 4 | 24 |  |
| Description on how the new functionality  will improve robot automation. | 7 |
| List and description of the parts to be  added. | 6 |
| A detailed budget allocation for the whole  upgrade. | 4 |
| Mention of where to buy these parts. | 3 |
| Products with similar technologies | Products identified using similar technologies; a full description provided on  what they are. | 4 | 15 |  |
| Explanation of how the mechanism  operates. | 8 |
| Compare the technology with the one used  in the GoPiGo. | 3 |
| Research on the GoPiGo | Research showing how the GoPiGo works; clear explanation of the core disciplines of  the GoPiGo. | 5 | 10 |  |
| A list of suggestions on how to improve the GoPiGo performance. | 5 |
| Calculation on battery usage | Clear calculations of the battery usage and  how long the GoPiGo will run for without stopping. | 10 | 10 |  |
| List of all the risks related to  the project | A list of all the risks involved with building a robot and the relevant ways to mitigate  them. | 10 | 10 |  |
| Penalties | Project resubmitted due to plagiarism  (−10% on next submission) | −10% | |  |
| **Total** | | **69** | |  |

Table of Contents

**Project Description5**

Cyber-Random-Rover5

**(Section A) Question 1 6**

Lighting system circuit with the GoPiGo 6

Lighting system circuit Status: OFF (Simulation screenshot)6

Lighting system circuit Status: ON (Simulation screenshot)6

Code for integrated LED lights circuit in the GoPiGo robot 8

**Question 2 9**

**AI Functionalities** **9**

The two functionalities would add9

Ultrasonic Sensors9

Infrared Sensors9

Estimated Budget9

**Products with Revers Memory Functionality10**

iRobot Roomba10

DJI Drones10

Comparison with my Cyber-Random-Rover (created with GoPiGo3)10

**Core Disciplines of GoPiGo10**

Mechanical Engineering10

Electrical Engineering10

Computer Science10

**Power Usage Calculation10**

Average Motor Power Consumption10

Calculations and Battery Duration10

**Potential Hazards and Safety Measures10**

Electrical Hazards10

Mechanical Hazards10

Battery Hazards10

**Robots Source Code (cyber\_rover\_v3.5.ipynb)10**

Import all necessary libraries & prepare live memory10

Get number of steps from user11

Prepare robot movements system & indication system12

Generate Random Movements & Save to CSV15

Create Control Buttons17

Restore & Run Movements from live memory18

Project Description

Cyber-Random-Rover

Cyber-Random-Rover in simple terms is a robot that uses a program that automates the GoPiGo with a memory that allows it to remember instructions. This robot drives around freely with no lines to follow, it does not just drive around in straight line but in random directions and straight lines for a while sometimes makes U-turns. The Cyber-Random-Rover has a memory so that it remembers the instructions that were processed. When the robot has finished processing all the movements it then performs all the movements backwards. The first instruction becomes the last instruction to be processed.

All this is made possible by the program I programmed using the Jupyter Notebook interface, which I access through the IP address 10.10.10.10 after connected to the Wi-Fi of DexterOS designed for the GoPiGo3.

These are the following files I use:

1. Python file (cyber\_rover.ipynb)
2. CSV file named (saved\_directions.csv)
3. Text file named (live\_memory.txt)

**Python file**

Cyber\_rover.ipynb is the backbone of the robot’s purpose. This file contains code that generates the random directions the robot wants to go; the user only specifies how many directions/steps the robot should take. Not only that but this file is also responsible for creating the csv and the text file.

**CSV file**

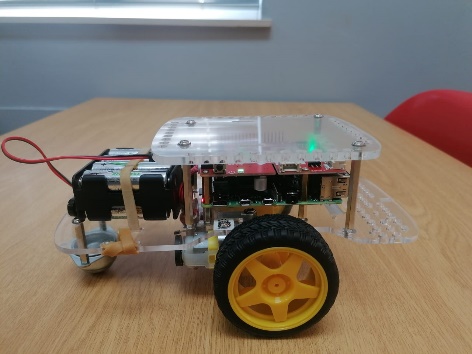
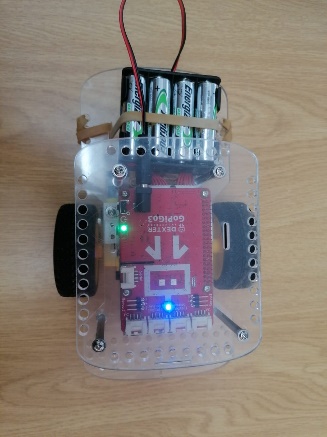
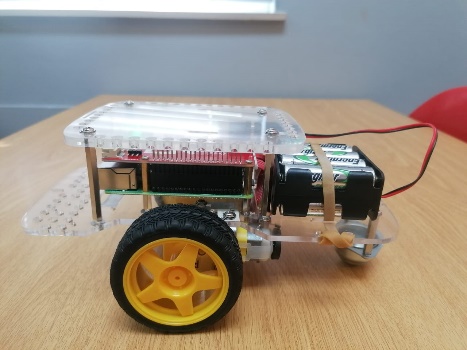
Saved\_directions.csv is responsible for saving the programs generated directions so that the python file can retrieve to run them on the robot or retrieve to reverse them then run them on the robot.

**Text file**

live\_memory.txt is the robot’s live memory. Its purpose is to act as the storage place for the python program. The robot’s activities must be consistently saved so that if the robot runs out of batteries during the run, next time it is started it needs to be able to start from its last instruction before it shuts down. I chose the text file because it is the most flexible file to store and retrieve and importantly to iterate through the file.

**GoPiGo3 Robot**

Robot’s Right Side Disconnected (Green) Connected (Blue) Robot’s Left Side

Section A

Question 1

1.4

a.) Create a fully labelled drawing of the circuit to integrate this lighting system with the GoPiGo. The circuit should be integrated with the Raspberry PI.

b.) Indicate the components that will have to be added in order to complete the project correctly.

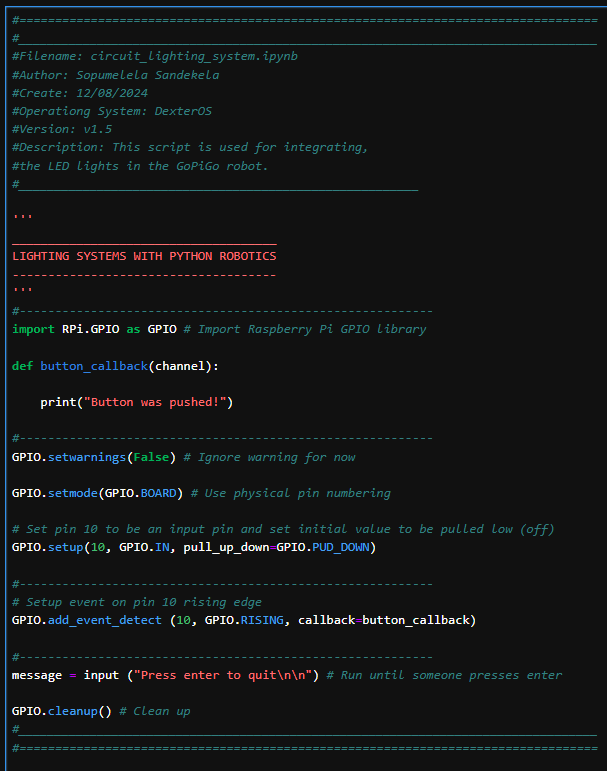
A circuit board with wires and wires

Description automatically generated

A diagram of a circuit board

Description automatically generated

1.5 Write down the code that will be used for integrating the LED lights in the GoPiGo robot.



Question 2

2.1)

## AI Functionalities

To improve the projects automation, it is important to implement AI functionalities. These functionalities aim is to help the robot work with obstacle detection and avoidance allows the robot to navigate around objects in its path, improving its autonomy and efficiency.

**The two functionalities I would add are:**

* Ultrasonic Sensors: These sensors detects and measures distance, a couple of for front and back.
* Infrared Sensors: These can help with proximity detection and object avoidance.

**Ultrasonic Sensors**

* Cost: Approximately R 100 - 200 per sensor.
* Source: Available at local electronics stores like RS Components as I’m located in South Africa, [Communica](https://www.communica.co.za/), or online platforms like [Takealot](https://www.takealot.com/).

**Infrared Sensors**

* Cost: Approximately R 40 - 80 per sensor.
* Source: Same as above.

**Estimated Budget**:

* 2 Ultrasonic Sensors: R 200 - 400
* **2** Infrared Sensors: R 80 - 160
* Total: R 280 – 560

2.2)

## Products with Revers Memory Functionality

**iRobot Roomba:**

* Mechanism: Roomba vacuums use infrared and bump sensors to create a map of the area they clean. They store this map in memory and can retrace their path if they need to pause and resume cleaning.
* Memory Saving and Reuse: The memory is saved in the device’s onboard storage and used to resume cleaning from the last point on interruption.
* Cost: Approximately R5,000 – 10,000 depending on the model.

**DJI Drones:**

* Mechanism: DJI drones use GPS and onboard sensors to create a map of their flight path. They can save this path and use it to return to the starting point or retrace their flight.
* Memory Saving and Reuse: Flight data is stored in the drone's internal memory and can be used to resume operations if interrupted.
* Cost: Approximately ZAR 10,000 - 30,000 depending on the model.

**Comparison with my Cyber-Random-Rover (created with GoPiGo3)**

* Technology Used: The GoPiGo Rover uses basic file storage to save instructions, whereas Roomba and DJI drones use sophisticated sensors and mapping algorithms.
* Memory Management: Roomba and DJI drones use onboard memory and advanced navigation systems, while the GoPiGo Rover uses CSV files for storing movements and text file as a live memory storage space.

2.3)

## Core Disciplines of GoPiGo

**1. Mechanical Engineering**

* Improvements: Upgrade the wheels or chassis for better durability and stability.
* Cost: Wheel upgrades can cost around R 200 - 400.

**2. Electrical Engineering**

* Improvements: Enhance the motor control system or add new sensors for better performance.
* Cost: Motor controllers or sensor upgrades can cost around R 300 - 800.

**3. Computer Science**

* Improvements: Develop more advanced control algorithms or integrate new AI functionalities.
* Cost: Software development is often a time investment, but hiring a developer could range from R 5,000 - 10,000.

2.4)

## Power Usage Calculation

The GoPiGo3 I have come with a base kit, Raspberry Pi (with installed Raspbian SD card) and 8 AA Alkaline AA batteries which can yield 2000 to 3000 milliampere-hours (mAh).

**Average Motor Power Consumption**:

* Voltage: 6V
* Current: 0.75A per motor

**Battery Capacity**: 2,000mAh (2Ah)

**Calculation**:

1. Total Power Consumption: 2 motors×0.75A = 1.5A
2. Battery Life: 2Ah ÷ 1.5A = 1.33 hours
3. In Minutes: 1.33 hours×60 minutes/hour = 80 minutes

**Battery Duration**: Approximately 80 minutes.

2.5)

## Potential Hazards and Safety Measures

**1. Electrical Hazards**

* Hazard: Risk of short circuits or electrical shocks.
* Safety Measures: Use insulated wires, avoid exposed connections, and implement circuit breakers.

**2. Mechanical Hazards**

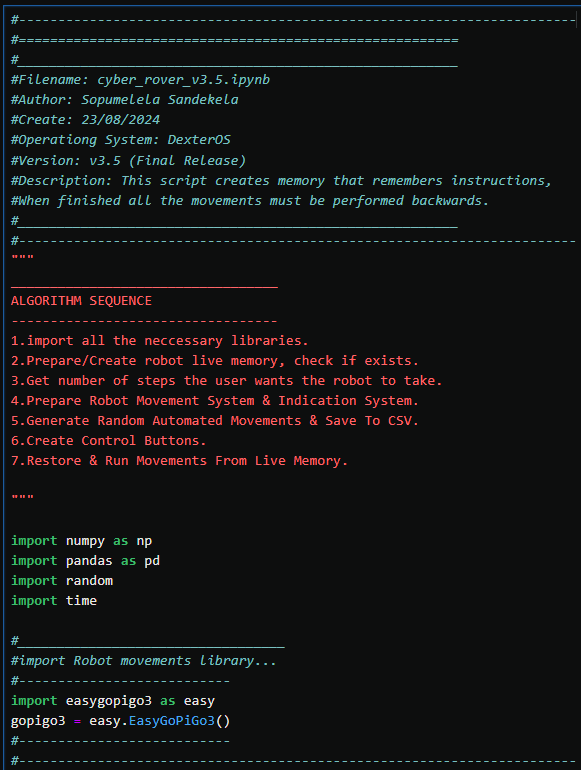
* Hazard: Injuries from moving parts.
* Safety Measures: Cover moving parts with protective shields and ensure safe operational environments.

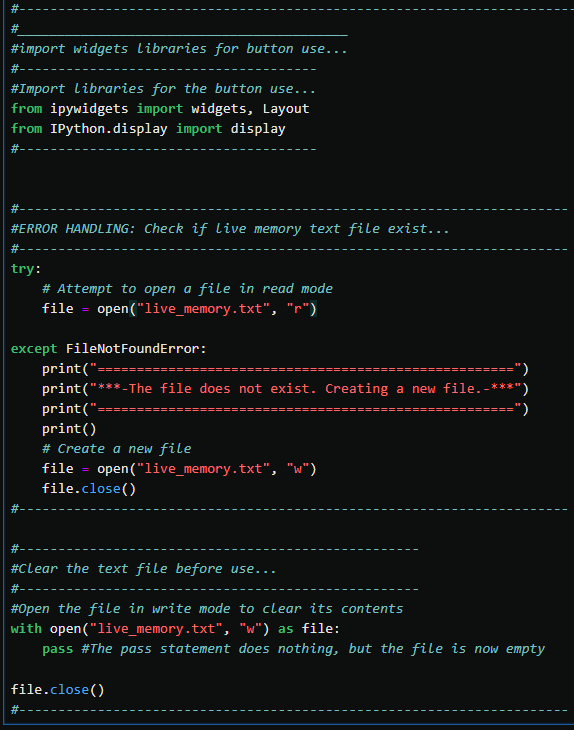
**3. Battery Hazards**

* Hazard: Battery leakage or overheating.
* Safety Measures: Use proper battery management systems and ensure adequate ventilation.

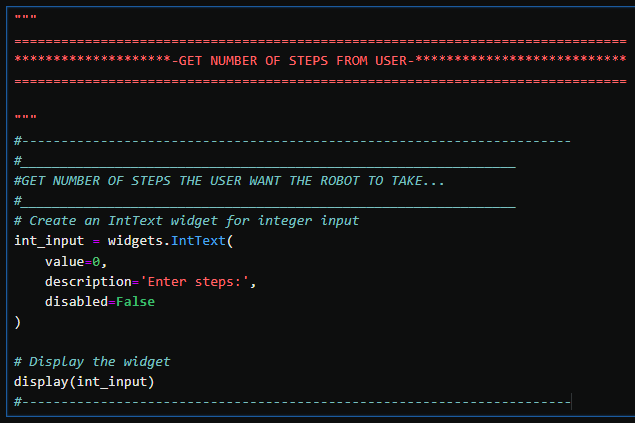
Source Code

## Import all necessary libraries & prepare Live Memory





## Get number of steps the Robot should take from User

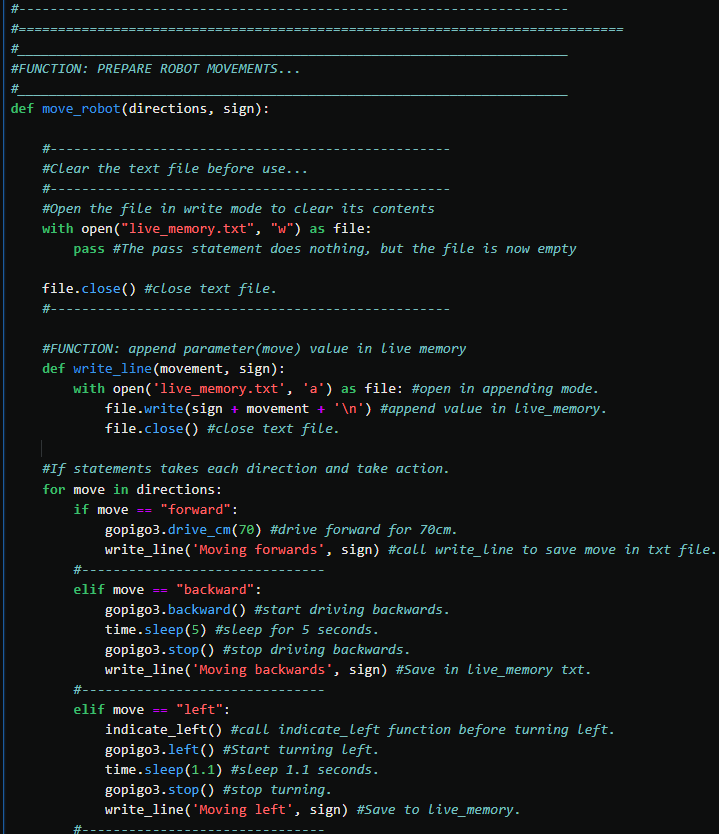


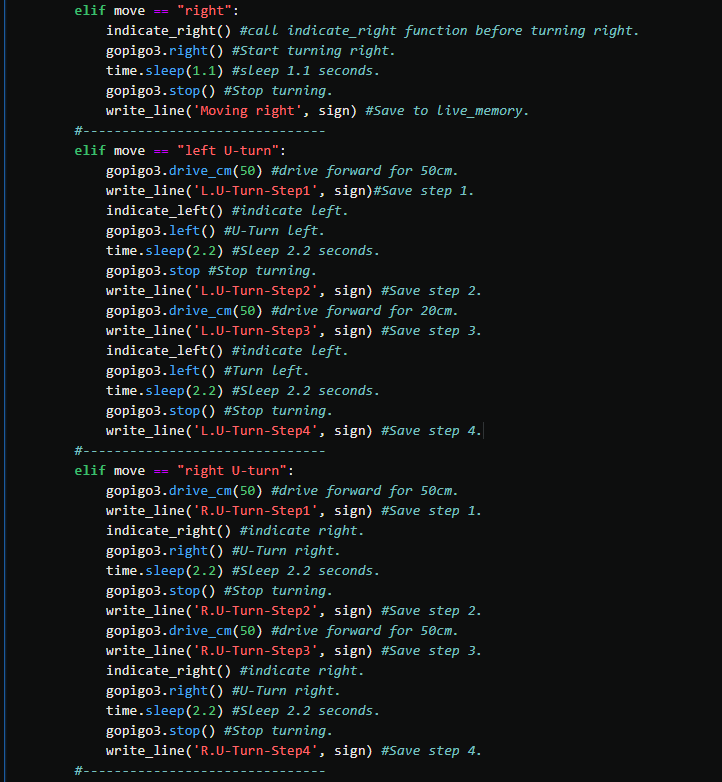
## Prepare robot Movements system & Indication system

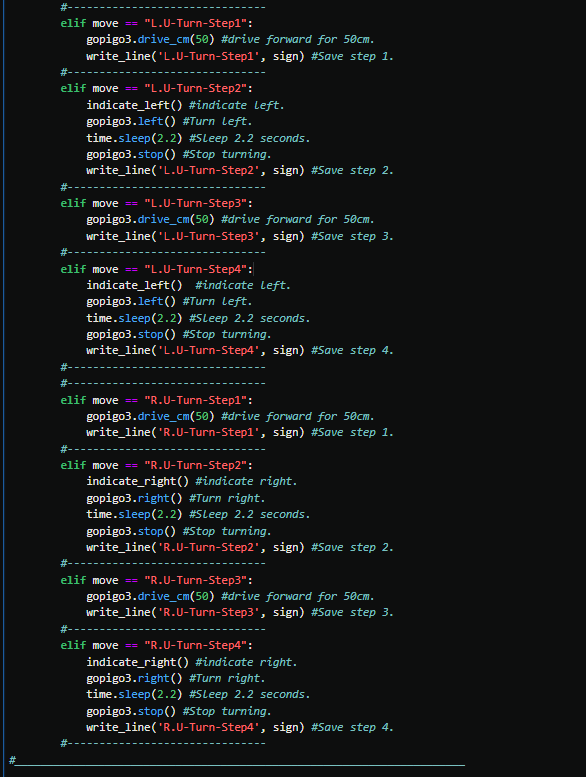
A screen shot of a computer program

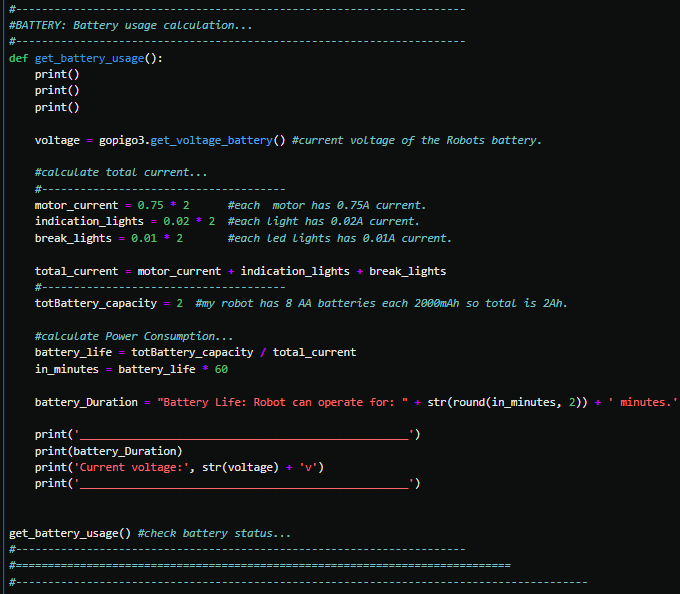
Description automatically generated



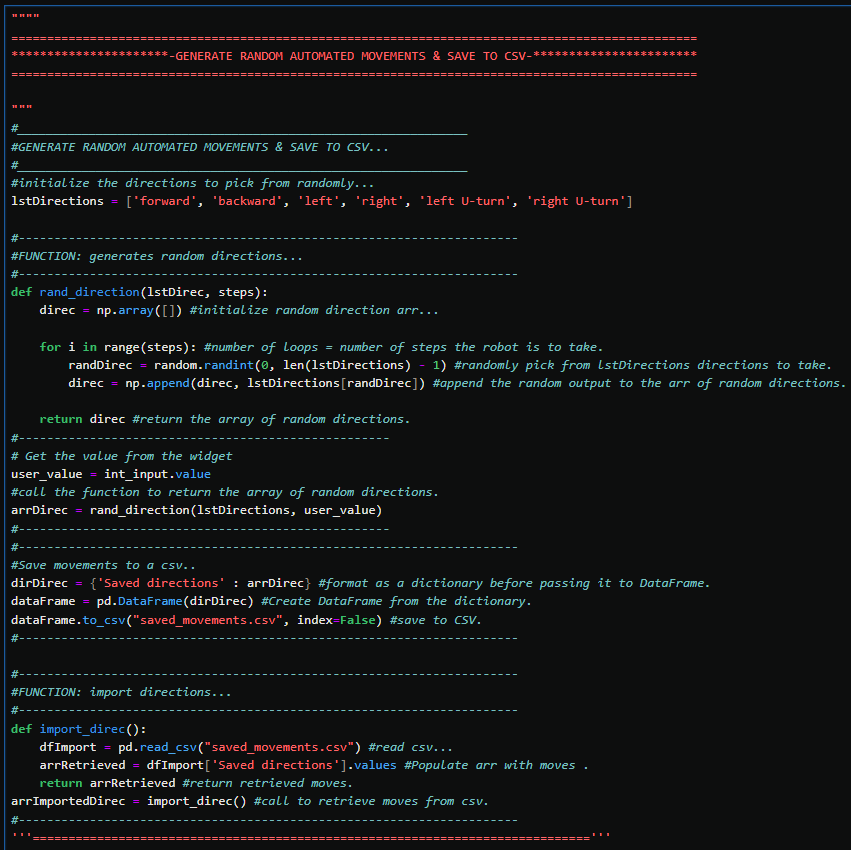








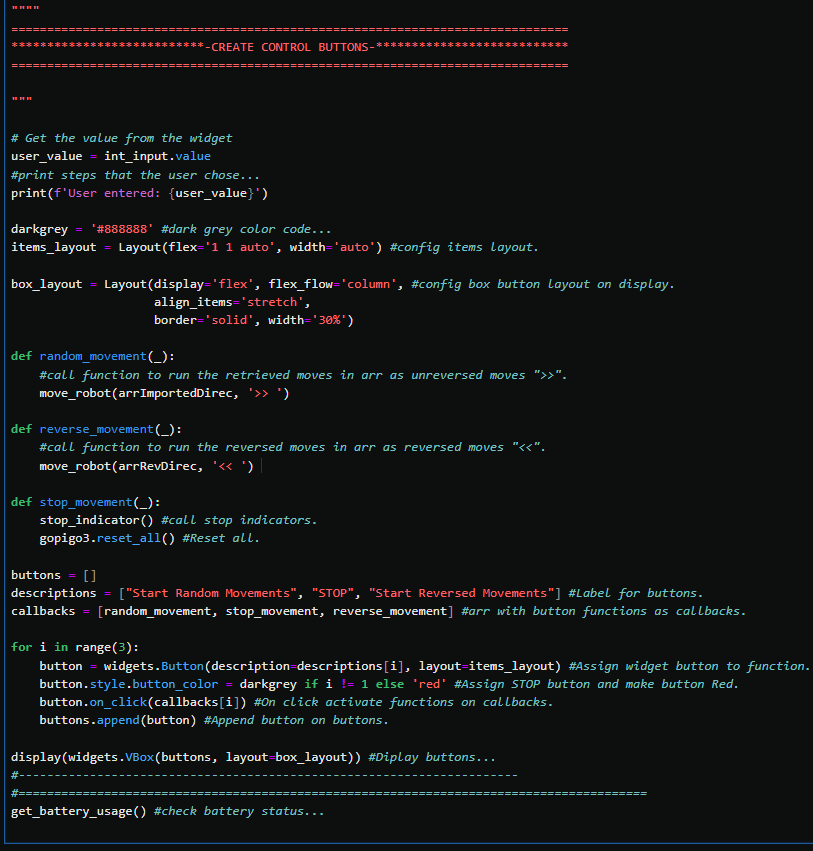
## Generate Random Automated Movements & Save to CSV



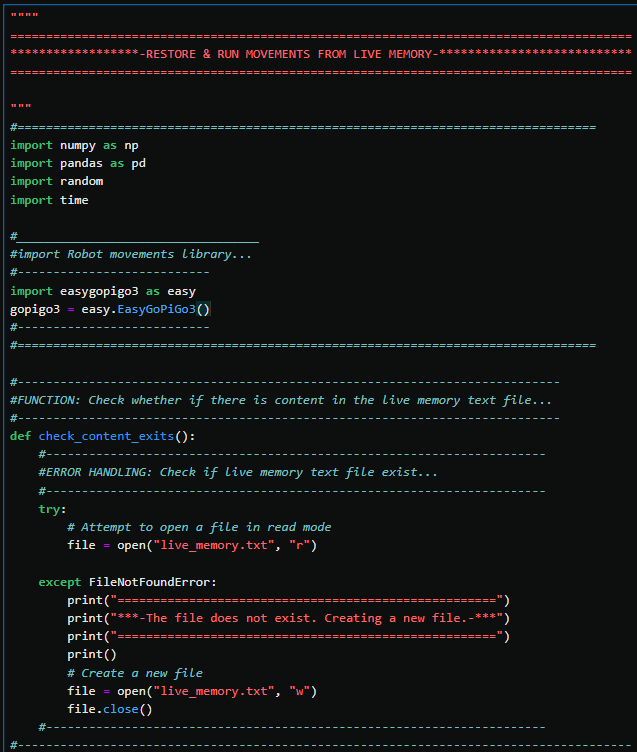
A black screen with white text

Description automatically generated

## Create Control Buttons



## Restore & Run Movements from Live Memory



A screen shot of a computer program

Description automatically generated

